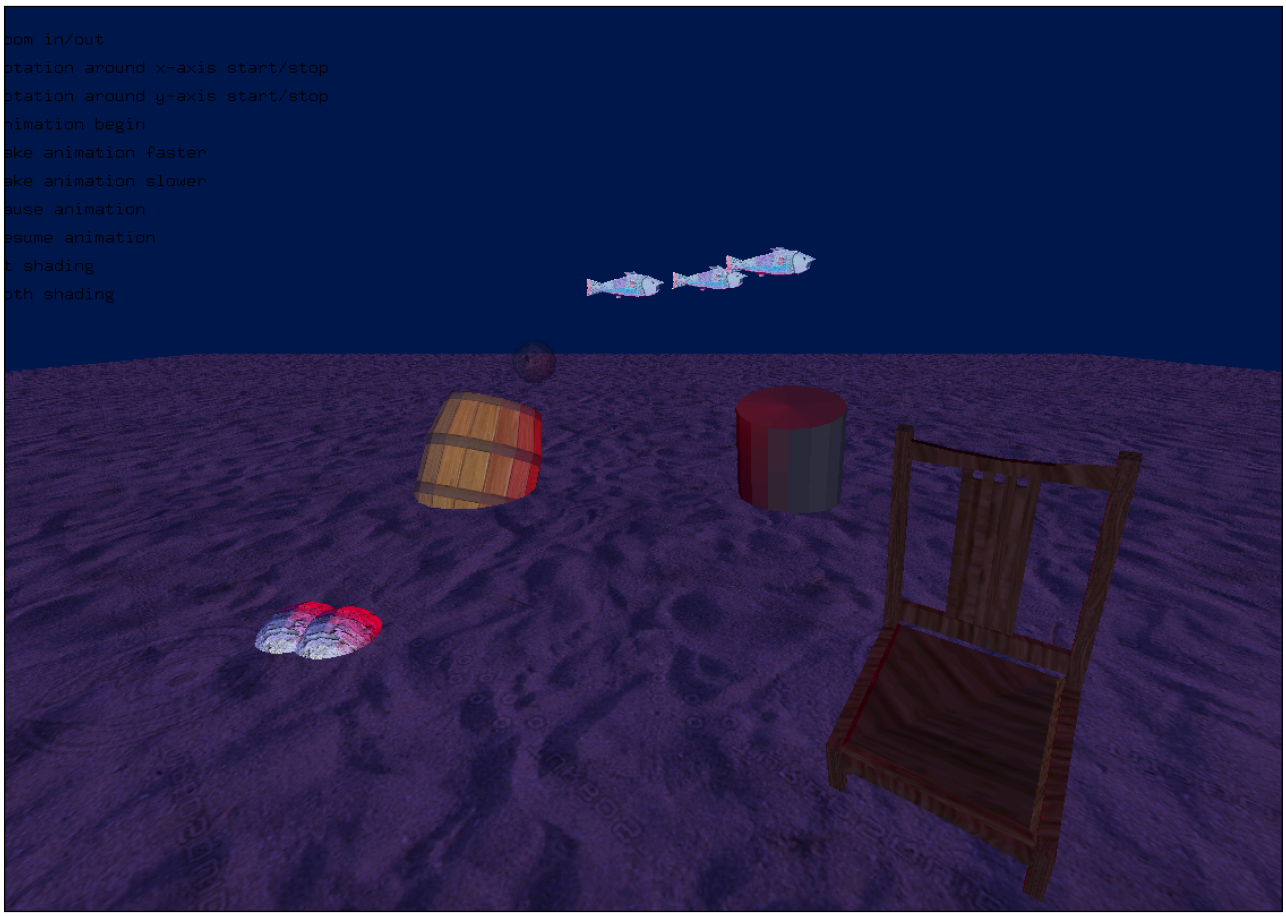
CG200 - OpenGL Assignment

by Campbell Pedersen

30 October 2016



Features Implemented

Zoom in/out and scene rotation

These are implemented as keyboard shortcuts, as described in the requirements. Zooming in is done by scaling the entire scene up/down, and scene rotation is done by rotating the entire scene about the x and y axes in a clockwise direction. Each keypress controls global values (_cameraAngleX, _cameraAngleY, _zoom), which are accessed in the camera setup subroutine. These are updated each frame, depending on the program's boolean state. Also, slices for glu objects like spheres are increased as zoom increases, and decreased as zoom decreases.

Flat shading/smooth shading

These are implemented as keyboard shortcuts, as described in the requirements. The P and p keys control a global boolean (_smooth). This is checked every frame in the camera setup subroutine, and smooth shading is applied to objects with the suitable normal vectors (glu objects are great for this, as their normal vectors are given).

Animation

The animation present includes fish swimming around the scene, and a bubble rising/falling near a wooden barrel. The keyboard shortcuts implemented control starting/pausing animation and the speed of the animation. The actual transformations to these objects is done while rendering them. These specific transformations depend on one global variable (_angle), which is changed only if the global animation boolean (_animating) is true. If false, _angle will remain unchanged and the animation will effectively be paused. If the animation is slowed, the frequency that this value is updated is reduced. The opposite applies to speeding up the animation. The fish use this variable to rotate about the Y axis, and the bubble translates on the Y axis in respect to the sin of the angle variable.

On screen text

The instructions for the keyboard bindings are rendered in 2D on the top left of the window. Instruction lines are stored in an array, where each line is written by using glutBitmapChar, and iterating through each string. After one line, the raster repositions to the line below and text continues to be written. All transformations are cleared before text rendering to ensure that the text does not transform with the scene.

Transparency

Transparency is implemented on the bubble by combining colour blending with the drawing colours alpha value. This ensures that whatever is behind the bubble is rendered properly.

Object Modelling/Loading from File

Some composite objects are loaded from files, in the form of .obj files, which are exported from Blender. Models were either made manually or found online (see references). The algorithm for reading such a file into a program can be found online (see references). These object files have not only their vertices loaded, but their texture co-ordinates and normals as well. These are all carried into vectors, where they're iterated through later and drawn as glVertexs, glNormals and glTexCoords. However, a few .obj export options are needed to properly parse the file. Materials are ignored, and each face is triangulated if not already. This increases the amount of faces overall, but improves code readability and makes the algorithm for rendering objects simpler.

Other objects were done using glu's methods, such as gluSphere, gluCylinder and gluDisc. These are great for smooth shading, as their normal vectors are already set up to allow for smooth shading.

Texture Mapping

Textures are loaded from file using a library found online, as part of an OpenGL tutorial series (see references). Textures are mapped to objects based on their texture co-ordinates. The ground is textured differently to other objects, in the sense that the texture repeats multiple times to give the illusion of a very large texture image. The rest of the objects just have different textures wrapped to them, depending on what they are. For glu objects, texture co-ordinates are already given.

Lighting

Both ambient and directed light exist in the scene. The ambient light is a blue colour, to give the illusion of being underwater. The red directed light just contrasted well with the scene, and gave the entire scene a sense of perspective. These lights are enabled at the very beginning of the scene, and initialised each frame to account for any changes in the scene (transformations etc).

Object Methodologies

Ground

The ground is simple a flat quad, textured and given a vertical normal. Extremely simple, very effective.

Wooden Barrel

The barrel is an .obj file loaded from Blender. The object's specific model/textures can be found in the references. This turned out really well, but it needed some serious tweaking in Blender. To get the correct file format for .obj files, the object needs to be UV unwrapped, and stripped of any materials. The texture is mapped to the object in OpenGL rather than in the .obj file. It was also made to be lower-poly by using a modifier, to improve performance.

Steel Barrel

This barrel is made of gluobjects, one gluCylinder and two gluDisks. Originally, this was just a gluCylinder. However, gluCylinders aren't closed by default. So instead, the cylinder was made and then disks were added to either end by rotating the sphere by 180 degrees and using the same radius/slices. This worked remarkably well, and supports the zoom quality feature.

Fish

This fish is another imported .obj file found online, loaded from Blender. Again, this needed to be UV unwrapped. However while doing this, I realised that animation would be MUCH easier to implement if the fish was not at the origin of the .obj. The texture worked remarkably well, given the already defined texture co-ordinates.

Chair

This is yet another imported .obj file. This object is the most complex of all of the objects, with many detailed parts. This is textured using a generic wooden look, instead of a texture meant specifically for the chair. This improves performance significantly, although the chair seems to still have a significant burden on the system's performance.

Rocks

The rocks are just gluSpheres, textured with a generic rock texture. These are translated below the ground, to give the illusion of them not being complete spheres. Also given their glu nature, zooming/smooth lighting works really well on these.

Bubble

The bubble is just another gluSphere, coloured with a lower alpha than 1 and blended with the scene to accomplish transparency. This carries the same benefits as all other glu objects.

References

Code

Image loader: http://www.videotutorialsrock.com/opengl_tutorial/textures/text.php
.obj Loader: http://www.opengl-tutorial.org/beginners-tutorials/tutorial-7-model-loading/

Models/Textures

Chair: http://opengameart.org/content/arts-and-crafts-chair-lowpoly
Barrel: http://opengameart.org/content/barrels-0
Fish: http://opengameart.org/content/basic-fish-low-poly